

TANK ATTACK!

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computadores
Algoritmos y Estructuras de Datos II (CE 2103)
II Semestre 2024

OBJETIVOS

GENERAL

- Diseñar un juego sencillo en C++ utilizando programación orientada a objetos

ESPECÍFICOS

- Aplicar conceptos de orientación a objetos
- Aplicar algoritmos de pathfinding
- Aplicar patrones de diseño en la solución de un problema.

REQUERIMIENTOS

El proyecto consiste en el diseño e implementación de un video juego retro llamado Tank Attack! A manera de ejemplo, dicho juego se ve de la siguiente forma:



El mapa consiste en una cuadrícula de un tamaño definido por lo estudiantes. La cuadrícula corresponde a un grafo implementado mediante matrices de adyacencia. Dicho mapa se genera aleatoriamente con obstáculos cada vez que se inicia el juego.

El juego permite que dos jugadores participen de forma local por turnos, utilizando el mouse. El juego coordina los turnos según corresponda. Gana el jugador que destruya los tanques del otro o el que tenga más tanques cuando el tiempo se acabe (máximo 5 minutos).

En su turno, el jugador puede únicamente mover un tanque, disparar o aplicar un power-up.

A cada jugador se le asignan cuatro tanques, dos de un color y dos de otro color. El color azul y el color celeste tienen funcionalidad equivalente. El color amarillo y el color rojo tienen funcionalidad equivalente. Rojo y azul son para el jugador 1; amarillo y celeste son para el jugador dos.

Cada jugador puede mover un tanque al seleccionarlo con el mouse y luego dar click en la posición a la que quiere ir. Se aplican las siguientes reglas:

1. Los **tanques celeste/azul** determinan la ruta mediante BFS o movimiento aleatorio, con un 50% de probabilidad entre un tipo de algoritmo u otro.
2. Los **tanques amarillo/rojo** determinan la ruta mediante Dijkstra o movimiento aleatorio, con una probabilidad de 80% de usar Dijkstra vs movimiento aleatorio.

Cada vez que un tanque determina la ruta, se muestra en pantalla un trazo de la ruta calculada. Dicho trazo se borra en el siguiente turno.

Al disparar, el jugador selecciona el tanque que va a disparar y hace click en el punto al que quiere atacar. Se aplican las siguientes reglas:

1. Normalmente, las balas realizan pathfinding de línea vista.
2. Las balas rebotan en las paredes con cierto ángulo definido por el estudiante.
3. La bala puede rebotarle al mismo tanque que la disparó y hacerle daño
4. Una bala que toque al tanque enemigo, realizar el siguiente daño:
 - a. 25% de daño a los tanques celeste/azul
 - b. 50% de daño a los tanques amarillo/celeste

Cada vez que un tanque dispara, se muestra en pantalla un trazo de la ruta calculada para la bala. Dicho trazo se borra en el siguiente turno.

En todo momento debe mostrarse el daño total de cada tanque.

Aleatoriamente, se generan power-ups para cada jugador. Un power-up consume un turno y se aplica al siguiente turno. Se tienen los siguientes power-ups:

1. **Doble turno:** se aplica en un turno y los siguientes dos turnos son para el mismo jugador.
2. **Precisión de movimiento:** aplica BFS o Dijkstra (según el color) con un 90% de probabilidad en el turno siguiente que sea de movimiento. Es decir, se consume el turno y al volver a ser el turno del jugador con el power-up aplicado, si hace un movimiento, se aplica con la probabilidad del power-up.
3. **Precisión de ataque:** la bala sigue el algoritmo A* hacia el objetivo seleccionado.
4. **Poder de ataque:** la bala tiene un poder destructivo de 100%

RUBRICA

REQUERIMIENTO	PUNTAJE
Modelado del mapa mediante grafos	10
Generación aleatoria del mapa	5
Multijugador	5
Control del tiempo del juego y regla general para ganador (al destruir todos o quien tenga más al finalizar el tiempo)	5
Funcionalidad de movimiento de los tanques	30
Balas	25
Power-ups	20

ASPECTOS OPERATIVOS

- El trabajo se realizará en **parejas**.
- El uso de Git y Github es obligatorio y se revisarán las contribuciones de cada miembro del equipo de trabajo.
- La nota es grupal, es decir, se dará una nota general y cada miembro del equipo indicará como se divide entre todos.
- La fecha de entrega será según lo especificado en el TEC Digital. Se entrega en el TEC digital, un archivo PDF con la documentación. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la cita revisión oficial.
- Cada miembro del equipo debe demostrar su contribución equitativa al proyecto, de lo contrario, queda a criterio del profesor revisar o no el proyecto.

DOCUMENTACIÓN

- La documentación deberá tener las partes estándar:
 - Portada
 - Introducción
 - Tabla de contenidos (con los títulos debidamente numerados)
 - Breve descripción del problema
 - Descripción de la solución
 - Por cada uno de los requerimientos, se deberá explicar cómo se implementó, alternativas consideradas, limitaciones, problemas encontrados y cualquier otro aspecto relevante.
 - Diseño general: diagrama de clases UML con las clases relevantes que muestren el diseño orientado a objetos y los patrones de diseño aplicados

EVALUACIÓN

- El proyecto tiene un valor de 20% de la nota del curso

- Los proyectos que no cumplan con los siguientes requisitos no serán revisados:
 - Toda la solución debe estar integrada
 - La interfaz de usuario debe estar implementada e integrada
- El código tendrá un valor total de 80%, la documentación 10% y la defensa 10%. De estas notas se calculará la *Nota Final del Proyecto*.
- Aun cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
 - **Si no se entrega documentación en formato PDF, automáticamente se obtiene una nota de 0.**
 - Si no se utiliza un manejador de código se obtiene una nota de 0.
 - Si la documentación no se entrega en la fecha indicada se obtiene una nota de 0.
 - El código debe desarrollarse en C++, si no, se obtendrá una nota de 0.
 - La nota de la documentación es proporcional a la completitud del código. Es decir, si el código no está completo, aunque la documentación lo esté, no se dará el total del 10%
- La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto.
- Cada estudiante tendrá 20 minutos para exponer su trabajo al profesor y defenderlo, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo que se recomienda tener todo listo antes de entrar a la defensa.
- Cada grupo es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberán avisar al menos 2 días antes de la revisión a el profesor para coordinar el préstamo de estos.
- Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.